CS 348: Computer Networks

- LLC-ARQ; 2nd Aug 2012

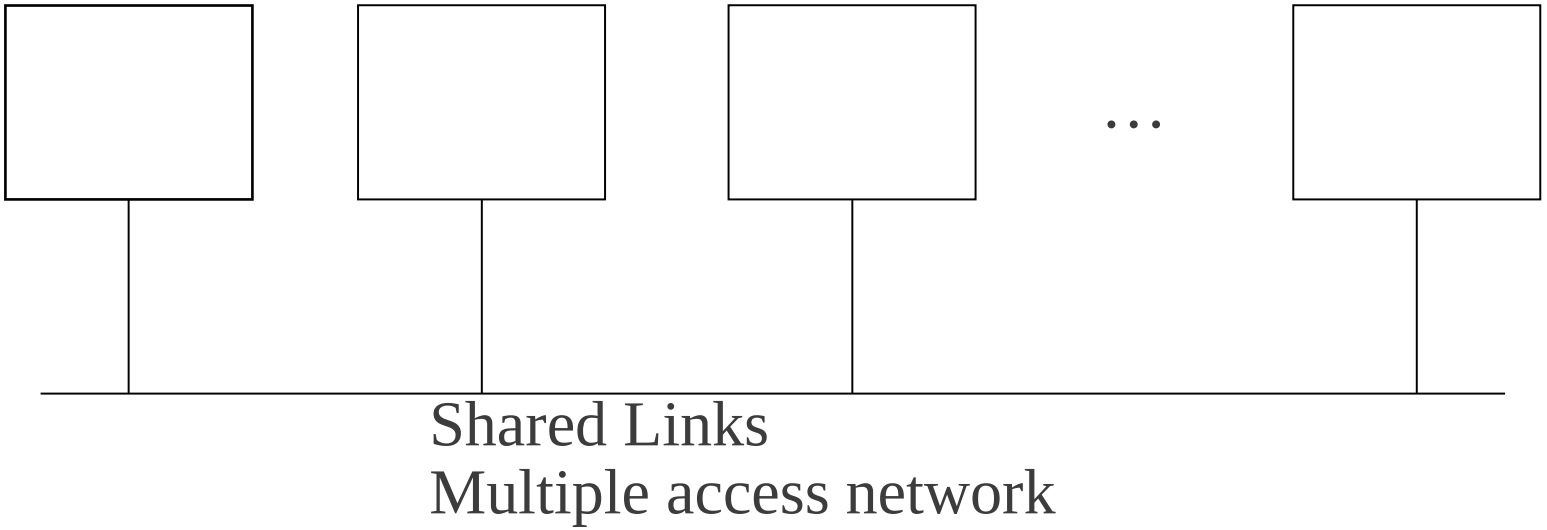
Instructor: Sridhar Iyer
IIT Bombay

Recall: Delay X Bandwidth Product

- Relative importance of bandwidth and delay
  - Small message: 1ms vs 100ms dominates 1Mbps vs 100Mbps
  - Large message: 1Mbps vs 100Mbps dominates 1ms vs 100ms
- Example:
  - 100ms delay and 45Mbps bandwidth

    => 560 KB of data in the pipe

We will move from a single link to ...



Today's class discussion

- Having seen the concepts in PHY, we will get into the Link layer (MAC and LLC), also known as Layer 2.
  - What should be the concerns of the Link layer?
  - What services should Link layer provide?
- Let us quickly put some answers on the board!

Data link layer

Controls a single physical link

Service interface to network layer

Data link functions

- Medium access control
  - Regulating multiple access to the medium
- Logical link control
  - Grouping of bits into frames
  - Dealing with transmission errors
  - Regulating the flow of frames

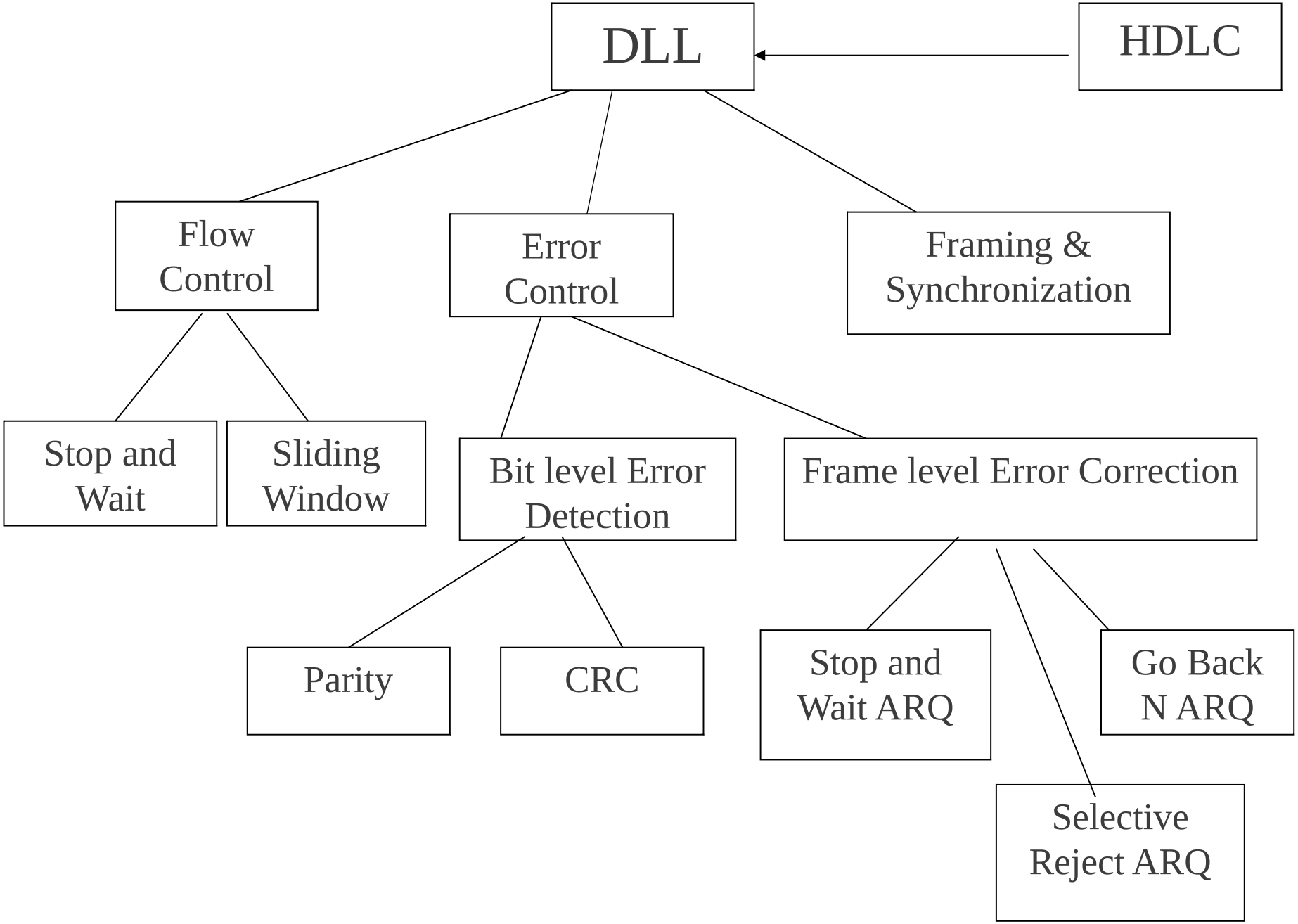
Logical link control (LLC)

- Framing (start and stop)
- Error Detection
- Error Correction
- Flow Control (optimal link usage)
- Examples: HDLC, LAP-D, PPP

Design Issues - ACKs

- Providing ACKs in this layer is an optimization not a requirement.
- Transport layer can very well provide reliable service

- Question to think about: In what cases should ACKs be implemented in link layer?


DLL
HDLC
Flow Control
Error Control
Framing & Synchronization
Stop and Wait
Sliding Window
Bit level Error Detection
Frame level Error Correction
Parity
CRC
Stop and Wait ARQ
Go Back N ARQ
Selective Reject ARQ

Example Frame Format : HDLC

Framing

- Starting and ending flags, with *bit stuffing*
  - each frame begins and ends with a special bit pattern, 01111110
  - allows data frames to contain arbitrary number of bits
  - allows character codes with arbitrary number of bits per character
- Self-study question: What is bit stuffing?

Bit level error detection

Single-bit, multi-bit or burst errors introduced due to channel noise.

- Detected using redundant information sent along with data.

- Full Redundancy:
  - Send everything twice
  - Simple but inefficient

Error detecting codes

- error-detecting codes: include enough redundancy to allow receiver to detect error
- more efficient and preferred solution
  - parity check
  - cyclic redundancy check (CRC or polynomial code)
  - checksum

- More on this later!

Frame level error correction

- Problems in transmitting a sequence of frames over a lossy link
  - frame damage, loss, reordering, duplication, insertion
- Two Solutions:
  - Forward Error Correction (FEC)
    - Use of redundancy for packet level error correction

Frame error & flow control - ARQ

- Problems in transmitting a sequence of frames over a lossy link
  - frame damage, loss, reordering, duplication, insertion

- Automatic Repeat Request (ARQ)
  - Detection: Sequence numbers, Timeouts
  - Correction: Use acknowledgements and retransmission

Sequence numbers

- In each header
- Incremented for non-retransmitted packets
- *Sequence space*
  - set of all possible sequence numbers
  - for a 3-bit seq #, space is {0,1,2,3,4,5,6,7}

Using sequence numbers

- gap in sequence space allows *receiver* to detect loss
  - e.g. received 0,1,2,5,6,7 => lost 3,4

- Receiver sends ACKs which carry *cumulative* seq #
- if no ACK for a while, *sender* suspects loss
  - need to choose timeout interval

Timeouts

- Set timer on sending a packet
- If timer goes off before ACK, resend

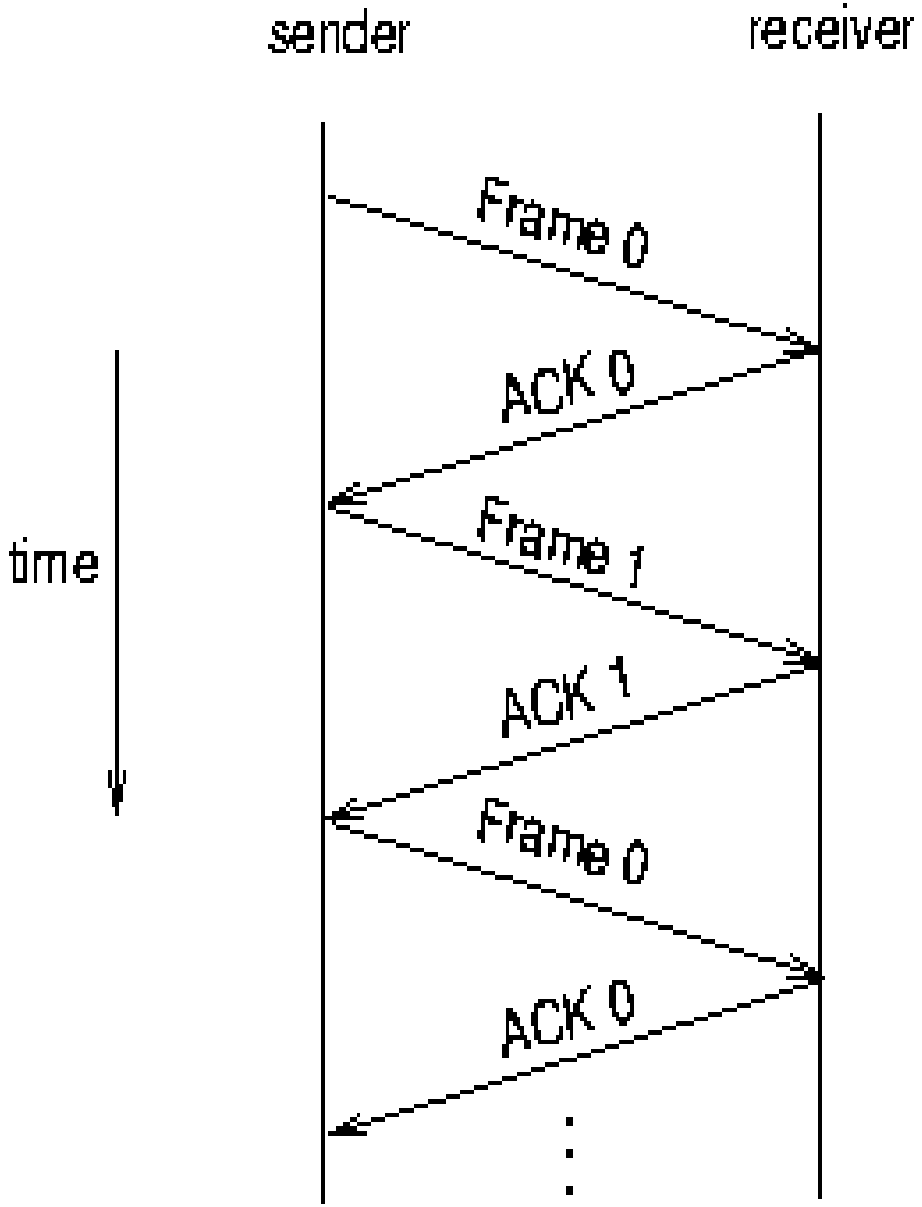
- How to choose timeout value?
- We expect a reply in about one round trip time (RTT)

Timeout schemes

- Static scheme
  - know RTT *a priori*
  - timer set to this value
  - works well when RTT changes little
- Dynamic scheme
  - measure RTT
  - timeout is a function of measured RTTs
- More on this later!

Retransmission Schemes
Stop and Wait ARQ

- Sender waits for ACK after transmitting each frame.
- Receiver sends ACK if received frame is error free.
- Sender retransmits frame if ACK not received before timer expires.



Stop and Wait ARQ

- Simple to implement but may waste bandwidth;
- Example: 1.5Mbps link * 45ms RTT = 67.5Kb (8KB).
  - Assuming frame size of 1KB,
  - stop-and-wait uses one-eighth of the link's capacity.
  - Sender should be able to transmit up to 8 frames before having to wait for an ACK.

Stop-and-Wait ARQ

1. View the link: http://oscar.iitb.ac.in/onsiteDocumentsDirectory/StopAndWaitARQ/StopAndWaitARQ/index.html

2. Play with the various settings in the animation till you are able to answer questions like: "What is the difference between the frame error case and ack error case?" There are also self-test questions at the end of the animation.